NMIS - Network Management Information System

Last updated 25 February 2002

NMIS Home

Table of Contents

- Introduction
- **Announcements**
- Sample Screens
- FAQ
- Latest Changes
- Contributors
- Contributions
- User Group
- Supported Platforms
- Browser Support
- Why?
- Features
- RRD Calculator **NEW**
- **Installation**
- **PATCHES**
- Apache Configuration
- Documentation
- Help
- Required Packages
- Downloads
- To Do
- GNU License
- NET-SNMP Notes

NMIS was originally written by Keith Sinclair, it is maintained by Keith and a collection of contributors around the world.

Download NMIS 2.00 Now!

NMIS is hosted by SINS - Sinclair InterNetworking Services

The software has been released to the Public Dom under the GPL License. This means that it is free i agree to the terms of the GPL License. If you woul like the software under a different agreement plea contact Keith Sinclair.

NB: NMIS is a hobby. Though some developmen continues this is limited by fulltime work and life commitments. I have released this software as I wanted to contribute something back to the Netwo Management Community which has helped me. C like Tobias Oetiker!

For more Cisco oriented NMS software try Cisco-centric Open Source Exchange Community.

NMIS assists with maintaining DNS LOC records RFC1876

Network Diagrams - Drawing network diagrams can be very difficult. How to represent complex network topologies easily and make them available for Network Operations and Management. This page discusses some strategies.

Introduction

NMIS stands for Network Management Information System. It is a Network Management System which performs multiple functions from the OSI Network Management Functional Areas, those being, Performanc Configuration, Fault.

It started as a SNMP polling and statistics viewer front-end to Tobi Oetiker's RRDTool. RRDTool replaces MRTG but doesn't include a front end and backend to handle SNMP polling and display resulting web page etc. The original NMIS evolved quite rapidly to meet demands of production environments. The backend, polling engine, uses SNMP to collect interface and health statistics for Cisco Routers, certain Cisco Catalys Switches and Generic SNMP devices every 5 minutes.

The backend stores the statistics in RRD's (Round Robin Databases) and ensures that devices are up, issu alerts, etc. The front end accesses the information stored in the RRD's and displays statistics the resulting graphs, reports, etc.

Both the front and back ends are highly extensible and features are easy to add as the structure is learnt. F example the backend was just collecting interface statistics every poll cycle, it was easy to add collection o health (cpu, memory, buffer, etc) and response time, availability.

I will continue to spell out more about NMIS but for now, have a look at the Features and Benefits, Install it

see what it can do.

NMIS is released as GPL software, please ensure that you read the license as by using this software you a bound by it.

ANNOUNCEMENTS

- NMIS 2.00 is available NOW!
- **NOTE: Lots of NMIS has been deprecated! Things that I wanted to do have been done, code simplified, filenames changed and lots of things. This has been done to improve and simplif NMIS but will be that if you have an existing NMIS running you will need to implement the ne NMIS carefully.**
- **NOTE: FOR ANYONE UPGRADING TO NMIS 2.0 ENSURE YOU CHECK ALL CONFIG ITEMS.**
- **NOTE: Changes now tracked in the Change Log.**
- With all the changes NMIS appears to be HEAPS faster!
- Many new config options and configuration items exist.
- Configuration validation routine for ensuring that NMIS is configured properly.
- Updated whole bunches of nmiscgi.pl, updated the style and revamped it a little bit, looks a bit bette now.
- Updated logs.pl with a better parsing algorithm, handles the weird messages in syslog which routers switches send.
- Added map.pl which enables NMIS to display group status colored icons for each group on a HTML using CSS. Icons will change color with group status, also display summary network metrics and all drill in to NMIS for each group. (not auto-discovery or drag and drop). Adds a bunch of config.dat elements and requires setting up map.csv with icon placement co-ordinates.
- Added summary.pl which is a very basic NMIS CGI interface for WAP and Palm type devices or low bandwidth connections.

Sample Screens - these "screen shots" are output from a production system but edited to protect the innocent. Some of the hyperlinks have been removed. A real system would have many hyperlinks to itself

NMIS has had many updates and changes since this sample was produced but this is still representative o what NMIS looks like and does.

Contributors - NMIS development has been assisted by (in no particular order): Wade Miller, Russell Mill Richard Kuehnle, Brian M Estep, Gary Veum, Dave Packham and Stephane Monnier

Contributions - NMIS was orginally developed by Keith Sinclair, since then quite a few people have contributed to its development. If you are interested in contributing to the development you are most welcc to just send your code, ideas, suggestions to the User Group.

User Group - To assist in keeping interested parties in sync I have created a NMIS user group on Yahoo Groups.

Post message:	nmis_users@yahoogroups.com
Subscribe:	nmis_users-subscribe@yahoogroups.com
Unsubscribe:	nmis_users-unsubscribe@yahoogroups.com
List owner:	nmis_users-owner@yahoogroups.com
URL for the user group:	http://groups.yahoo.com/group/nmis_users

Supported Platforms - NMIS is written in Perl so theoretically if your toaster runs Perl and has an IP addr can run NMIS. Nearly all the development of NMIS has taken place on Solaris for Sparc and i386. NMIS should work on all versions of Unix, and if you have any feedback on how the install went or problems plea let the team know @ nmis_users@yahoogroups.com so the team can catch the comments and make som more permanent changes for future releases.

NMIS is often run on Linux PC platforms but has also been run on higher end Sun Microsystems machines Memory is more important then CPU in this case. Performance for polling is dependant on many things bu usually limited by how long it actually takes to poll each device for the required SNMP statistics. Generally speaking any Pentium II or UltraSparc should be able to manage about 100 to 200 nodes. The faster the machine the more memory should equal more nodes. This will obviously reach some sort of law of diminis returns.

Recommended hardware for 100 nodes (highly conservative, would probably do a few more):

- Solaris I386 or Linux (or anything you want to support)
- Pentium III 800 Mhz
- 256 Megs of Memory
- 20 Gig of disk (cheap)
- Fast Ethernet connection to the network.

Browser - The CGI scripts of NMIS have been written with HTML 4.01 and CSS2 W3C recommendations mind. NMIS in general relies heavily on Cascading Style Sheets (CSS), so some of the older browsers do work quite write, for example Netscape 4.x do not render the web pages properly and look quite terrible. N has been tested with the following Windows browsers:

- Internet Explorer 5.x
- Netscape 6.x
- Opera 5.11

NMIS attempts to be fully validated HTML but does not promise to comply completely at present due to the large amount of HTML embedded in the CGI scripts.

Why?

Why write yet another piece of Network Management Software? Simple, most Network Management Soft does one thing, focuses on an element from the OSI Network Mangement Reference Model. This means t you end up with lots of different bits of software running, all of them polling the network and they all have to integrated. NMIS is trying to meet the requirements of several functional areas from the OSI network management model.

At the moment this is Performance Management and Fault Management and a little Configuration Management. That is why NMIS was written, to fill in the gaps and provide one system which will meet a l number requirements and functions of network management. NMIS is also aiming to be proactive, provide reporting and give the status of the network "at a glance fashion".

The at a glance is a dashboard of the network with operational status of all network devices and the groups which those devices belong to. NMIS could be considered an API for network management, get the polling engine to collect and monitor, using RRDTool as the database, then access this information in any way you to display status, statistics, etc.

Required Packages

The following packages and versions are what NMIS was built on, it is highly likely that other versions of th same programs will work but I have not performed the regression tests.

The following MOD needs to be made to SNMP_MIB sub_loadoids.html this code can be added to the SNMP_MIB.pm file in the SNMP Simple Package.

Package	Version Tested	Author	Download
Perl	5.6.0	Larry Wall	http://www.cpan.org/ports/index.html
RRDTool	1.0.33	Tobi Oetiker	http://ee-staff.ethz.ch/~oetiker/webtools/rrdtool/download.html
SNMP Session	0.77	Simon Leinen	ftp://ftp.switch.ch/software/sources/network/snmp/perl/
SNMP Simple		Alan Nichols	http://www.sins.com.au/public/SNMP_Simple.tar.gz http://www.sins.com.au/public/SNMP_Simple_NMIS.tar.gz (already patched)
Time::ParseDate	99.111701	David Muir Sharnoff	Time-modules-99.111701.tar.gz
Time::HiRes	1.20	Douglas E. Wegscheid	Time-HiRes-01.20.tar.gz
Apache	1.3.11	Apache Project	http://www.apache.org/dist/
Image Info	0.04	Gisle Aas	http://www.perl.com/CPAN/authors/id/GAAS/Image-Info-0.04.ta
NET-SNMP		NET-SNMP Project	NET-SNMP is not required for NMIS, it is however a VERY han tool for Network Mangement. http://sourceforge.net/project/showfiles.php?group_id=12694

Downloads

nmis-2-00.tar.gz	NMIS 2.00
nmis-1-00b.tar.gz	NMIS 1.00b
nmis-1-00a.tar.gz	NMIS 1.00a
nmis-0-9a.tar.gz	NMIS 0.9a
nmis-0-8c.tar.gz	NMIS 0.8c
nmis-0-8b.tar.gz	NMIS 0.8b
SNMP_Simple_NMIS.tar.gz	SNMP Simple NMIS Edited version
mibs.tar.gz	MIBs - A collection of publicly available MIBs sort of ready to use in general a with NET-SNMP.

NMIS - Installation

Last updated 21 June 2001	**Online Version**	**NMIS Home Page**

Installation

The packages listed in the required packages first following the instructions for each of these packages, some of these packages will have there own dependencies.

After the required packages are installed you will need to unpack (if you already haven't) the NMIS distribution. The base directory <BASE> is up to you a Solaris way is to use /opt/<PROGRAM> like /opt/nmis. I use /data/nmis myself. The file system which has the database is going to need a lot of space. You can have the program and library files stored in one file system or directory and the data files stored in another directory.

File locations are controlled with the config.dat file, this is the link for the program files to read your configuration you can modify the source to have a different config file but all other directories and control files are stored in this files the default is /data/nmis/files/config.dat

Extract the distribution from the compressed tar file

Uncompress the distribution

```
gunzip <nmis distribution gz file>
```

Pick a base directory for the NMIS distribution like /data or /opt and untar the files

```
tar xvf <nmis distribution tar file>
```

Directory Structure

The basic directory structure is:

Directory	Required Size	Type	Description
<BASE>/bin	small	Program	directory for all program files - these will change with later releases
<BASE>/cgi-bin	small	Program	directory for CGI program files - these will change with later releases
<BASE>/conf	small	Program	configuration files and some data files - these won't change
<BASE>/lib	small	Proram	library files - these will change
<BASE>/mibs	small	Program	mib files used by the program - these may or may not change
<BASE>/htdocs	medium	Data	files which are generated for the web pages
<BASE>/var	medium	Data	system files - generated
<BASE>/database	LARGE	Data	database files are stored here - generated
<BASE>/logs	medium	Data	log files

At the moment you should make all directories listed in this file, might automate this later on. Might be a good idea to decide on a location for the data files and keep it separate from the program files as you are likely to upgrade to new program versions.

Create nmis users and groups

Not a bad idea to have someone own nmis from the unix permissions point of view, I create a NMIS user and group and put my own userid the HTTPD user and whoever else in the group of nmis and then administration becomes a little easier. I have already set the distribution to be owned by userid 4200 and groupid 4200.

For Solaris:

```
groupadd -g 4200 nmis
useradd -u 4200 -g nmis -c "NMIS User" nmis
```

Setting the location of Perl in the scripts

So that NMIS runs properly you will need to edit all the perl scripts and change the top line which tells the script which shell to use currently this line would read:

```
#!/usr/local/bin/perl
```

This should work for 99% of people as this is the usuall location of Perl. If your executable lives somewhere else then you would need to change this ie:

```
#!/usr/bin/perl5
```

This should be done in nmis.pl, nmiscgi.pl, reports.pl, logs.pl and admin.pl.

Setting the location of library files in the scripts

If you have used the standard directories nmis should find all its own config files, about the only thing to do is to make sure NMIS can find rrdtool.

NMIS uses this for the rrdtool libraries:

```
use lib "/usr/local/rrdtool/lib/perl";
```

You might have installed rrdtool to the following directory like /usr/local/rrdtool/rrdtool-1.0.33. The easiest thing to do is create a symbolic link like this:

```
ln -s /usr/local/rrdtool-1.0.33 /usr/local/rrdtool
```

If you do this, as you upgrade rrdtool you can easily change the symbolic link without having to edit all the NMIS scripts.

Fixing File Locations

NMIS was written on Solaris so until version 2.2 is finalised there are some hard-coded file locations you might have to change in /bin/admin.pl change the file locations for ping, traceroute etc. to suit your machine (the command 'which ping' will tell you where it lives). You may need to change the syntax of ping - for eample the following works in Linux at about line 89 of /cgi-bin/admin.pl

```
if ($admin eq "ping") {
                $adminoutput=`/bin/ping -c 5 -s 64 $node`;
        }
```

The same applies for other commands. This will be automagically fixed in 2.2 :)

Also change the first line of run-reports.sh to #!/bin/sh if you don't have ksh at /bin/ksh ('which ksh' will tell you where it is or report an error if you don't have it at all.)

Setting up Ping

Ping is now Perl Net::Ping it requires NMIS to run as root to use the Net::Ping module. You could also try setting the sticky bit to allow nmis to run with root privilages.

Adding devices to the device list

The nodes.dat file lists all the devices you want to manage, this is a CSV file where each line defines node,community,net,type,role,group,collect.

Where:

- **node = The Node**
- **community = SNMP Community String**
- **net = Network Type lan or wan**
- **node type = switch or router**
- **role = core, distribution or access**
- **group = Nominal Location Group or the like**
- **collect = true,false do or don't do stats collection**

Populating the Locations and Contacts Tables

In the directory <nmis base dir>/conf/ copy the locations-sample.csv and contacts-sample.csv to locations.csv and contacts.csv respectively. The edit these files and create entries for each Contact and Location in your network, the files are TAB delimited so you can use comma's in the data fields. This will be linked to the device SNMP sysLocation and sysContact, so NMIS will allow you to lookup this information dynamically.

Change the default location and contact parameters especially the default contact emails address to whatever you want the notifications to go.

Configuring NMIS with the nmis.conf file.

If you are opting for a default install directories then you shouldn't have to change much. The default directory is /usr/local/nmis and the only thing you might choose to do is to put the database files somewhere else ie /bigfilesystem/nmis/database for example.

Some entries you will have to change are :

- **domain=sins.com.au.**
- **nmis_host=www.sins.com.au**

Some entries you might like to change are :

- **dash_title=NMIS Dashboard**
- **show_non_collected_interfaces=true**
- **show_large_menu=true**

Adding links to the links list

More on this later, working on generating a list of Point to Point Links automatically. This is REAL close now.

Running NMIS

You can turn on debugging with debug=<true|false|0-9> for more information while running NMIS.

When run without debug, NMIS is silent, it will only report BAD things and log non fatals to the NMIS log -> nmis.log in the logs directory is the default.

First check that the config is all right!
<nmis base dir>/bin/nmis.pl type=config

Then run an update!
<nmis base dir>/bin/nmis.pl type=update

Then run collect! Minimum! Runs collection for all nodes in the node list.
<nmis base dir>/bin/nmis.pl type=collect

Will run NMIS for router only doing a health collection and a interface collection with debuging on. Very handy for figuring out problems.
<nmis base dir>/bin/nmis.pl type=collect node=router debug=true

You can look at error messages in the file <nmis base dir>/log/nmis.log

Setting up web access

To have NMIS produce a sample Apache config run:
<nmis base dir>/bin/nmis.pl type=apache

Apache should already be working, easiest thing to do is add aliases for the relevant directories ie:

```
Alias /nmis/ "<BASE>/web/"
ScriptAlias /cgi-nmis/ "<BASE>/cgi-bin/"
```

Add these entries to the apache configuration generally /usr/local/apache/conf/httpd.conf and restart apache either with the command or a kill -HUP on the daemon PID. Keep in mind file permissions for the HTTPD daemon and NMIS, I usually create a group called nmis and put myself and the HTTPD user into it.

Now all the CGI scripts and base docs and style sheets will be linked in. You should be able to see something at http://<HOST>/cgi-nmis/nmiscgi.pl and wallah you should have something there.

Event Management and SNMP Tools

The NET-SNMP nee UCD-SNMP has a bunch of great SNMP tools, obtaining and setting up this package compliments NMIS quite well. Some work is being done to investigate the use of the NET-SNMP Perl SNMP module in place of the current SNMP API. More info on NET-SNMP and setting it up is here.

NET-SNMP IS NOT REQUIRED TO RUN NMIS

Setting up SYSLOG for Cisco SYSLOG

Add the following entry to syslog

```
local7.debug /var/log/cisco.log
```

If the log matches the one in logs.pl you will be able to browse the cisco syslog messages and see what is happening.

Setting up LOG

Log is a tool which allows viewing of the logs which NMIS and syslog generate. It color codes and allows searching, filtering of the log files.

The config file is /data/files/logs.dat, A sample is included in the NMIS distribution.

Automate the whole lot with CRON

Add the following entries to run NMIS bits and pieces.

```
#######################################################
# Run the Reports Weekly Monthly Daily
#######################################################
54 23 * * * /data/nmis/bin/run-reports.sh day health
55 23 * * 0 /data/nmis/bin/run-reports.sh week health
5 0 1 * * /data/nmis/bin/run-reports.sh month health
0 18 * * * /data/nmis/bin/run-reports.sh day response
57 23 * * 0 /data/nmis/bin/run-reports.sh week outage
#######################################################
# Run Statistics Collection
5,10,15,20,25,30,35,40,45,50,55 * * * * /data/nmis/bin/nmis.pl type=collect
#######################################################
# Run the update twice a day
30 7,20 * * * /data/nmis/bin/nmis.pl type=update
# Run the interfaces once and hour with Thresholding on!!!
0,15,30,45 * * * * /data/nmis/bin/nmis.pl type=threshold
```

NMIS FAQ - Frequently Asked Questions

Last updated 10 January, 2002	Online Version	NMIS Home Page

NMIS FAQ Maintained by Greg Ferro gferro@gfactor.com.au

- NMIS Home Page
- FAQ - Introduction
- Tips
- File size
- What is the format for the interfaces file?
- There are problems with the loadoids_file subroutine?
- How do I add MIBS to NMIS?
- Does NMIS support my network equipment ?
- I am having problems with Ping?
- Why do my Catalyst (4000, 5000, 6000) Switch Interfaces not appear in NMIS?
- Can't monitor my Linux - why?
- Do I need to do anything to SNMP Simple ?
- What does SNMP Simple do ?
- What does RRDtool do ?
- Does NMIS work on Microsoft Windows ?
- Why Don't ping and traceroute work?
- Why Aren't I getting any stored reports?
- How is the Group Summary metric derived ?
- What are the weighting for the Health Metric ?
- Can NMIS handle SNMP traps and syslog?

NET-SNMP IS NOT REQUIRED TO RUN NMIS BUT IS REAL HANDY

FAQ Introduction

- This FAQ is built from the combined musings of the many good folks on the NMIS User http://groups.yahoo.com/group/nmis_users mailing list . When a question comes up more than once try to put it in here. If your question is not answered here, then the mailing list is your next best optio
- The FAQ is maintained erratically and is completely arbitrary in nature. This means that you can sen me things and we can put them in.
- I can be contacted on gferro@gfactor.com.au
- If you have documented your install, please send us a copy, it would sure help a lot of people.

Tips: A couple of things to check before you start to panic !!

- Always check your file permissions, the files should be in the right group and the HTTP Daemon sho be able to read the files as well as write some directories.
- Did you check your config against the nmis-sample.conf?
- Did you run "nmis.pl type=config" to make sure the config was right?
- Have you run a "nmis.pl type=update debug=true"?
- Have you been running NMIS for more then 1 hour?

File Sizes:

NMIS creates lots of rrd's Round Robin Databases for each device it manages. The current statistics optio make the file sizes as follows:

- health rrd 2.5megabytes
- reachability rrd 756kilobytes
- interface rrd 568kilobytes.

All of these RRD's make up the database and for a few routers could become quite large quite quick.

Q. What is the format for the "interfaces" file?

A. The interfaces file is automatically generated by the NMIS system. You don't need to manually make the interface files. There is also an interfaces.dat which is all the interface data for all the devices put together also generated automatically by the update process.

Q. There are problems with the loadoids_file subroutine?

A. This is a modification to the SNMP Simple package, SINS has made up a distribution of SNMP_Simple availble here SNMP_Simple_NMIS.tar.gz

Q. How do I add support for my "name of box" to NMIS ?

A. NMIS was written to support Cisco routers and switches. Well, thats not quite true. NMIS can and does the standard MIB-II SNMP statistics that are available on modern equipment. So if you want to monitor a server, printer or some other brand of equipment just identify it as a "server" in the nodes.csv.

Of course, you aren't going to get the cool graphs that NMIS does for CPU, Memory or Buffers. This is because they are not standard info and moust be hardcoded into the Perl code. So if you want to monitor a Nortel router or a Cabletron switch you will have to get the MIBs and read which ones you want, then code NMIS. This is not difficult but you do need some Perl skills and you would have to be able to read a MIB an generate an OID file. (Of course, networking knowledge would also be required to configure the SNMP age on the device).

Q. How do I add mibs to NMIS?

A. You need to edit the loadmibs.pl script, put the names of the mibs into the script and generate the OID files. Not straight forward if you don't program in Perl but quite easy really. Something like:

```
mibdump.pl mibdir=/usr/local/share/snmp/mibs mibs=router outfile=/data/nmis/mibs/cisco-router.
```

You will need copies of MIBS, some available at http://www.sins.com.au/public/

Q. I am having problems with Ping?

A. NMIS now uses Perl Net::Ping and requires NMIS to be run with Root privileges. For backwards compatibility the old routines are included which execute Unix Ping.

Q, Why do my Catalyst (4000, 5000, 6000) Switch Interfaces not appear in NMIS?

A. NMIS requires a description in the configuration on the switch interfaces before it will collect data on the

This is a good idea. You normally only want to monitor ports that connect to servers or uplinks and who wa to spend time unmarking the interfaces you don't want. Far better to put a description and let NMIS build th config dynamically, than to spend time selecting which one of the {large number} of interfaces you want to monitor (a la MRTG when you use configmaker).

Q.I have a Linux server that I would like to test reachability of, and preferably graph interface statistics for. got the ucd snmp server on it, and an snmpwalk from the box nmis is running on to it gives me all the information I would think would be needed.

If I run an update or collect with debug true, it always tells me: "No system file exists creating one."

and once I'm done, neither <BASE>/var/$node.dat nor <BASE>/var/$node-interface.dat exists. This is the case when the node type is server or router.

Also, <BASE>/database/health/server/$node-reach.rrd exists, but with no data in it, and <BASE>/database/interface/server/$node/ exists, but there are no files in it.

A. "The main thing is the removal of the check on system.sysServices, as ucd-snmp doesn't define this (so NMIS will always think that SNMP is not working for Linux boxes). "

Q. Do I need to do anything to SNMP Simple ?

A. No. SNMP Simple is now a part of the NMIS files and is already in place. You can ignore those files.

You should take a few seconds to have a look through it and familiarise yourself with what it does though. I not compulsory just a good idea.

Q. What does SNMP Simple do ?

A. SNMP Simple is a set of Perl 5 modules that perform SNMP access requests. They written by Simon Leinen simon@switch.ch. Check his website for full information. The SNMP Simple package contains Perl modules SNMP_Session.pm and BER.pm, which, when used together, provide rudimentary access to rem SNMP (v1/v2) agent.

This module differs from existing SNMP packages in that it is completely stand-alone, i.e. you don't need to have another SNMP package such as NET-SNMP (as other monitoring packages do). It is also written enti in Perl, so you don't have to compile any C modules (very handy and easy to use). It uses the Perl 5 Socket.pm module and should therefore be very portable, even to non-Unix systems (so therefore works o unusual platforms like Amiga, Alpha and even Microsoft Windows).

The SNMP operations currently supported are "get", "get-next", "get-bulk" and "set", as well as trap genera and reception. NMIS doesn't use the trap features at this time.

Q. What does RRDtool do ?

A. RRDtool refers to Round Robin Database tool. Round robin is a technique that works with a fixed amou data, and a pointer to the current element. Think of a circle with some dots plotted on the edge, these dots the places where data can be stored. Draw an arrow from the center of the circle to one of the dots, this is pointer. When the current data is read or written, the pointer moves to the next element. As we are on a cir there is no beginning nor an end, you can go on and on. After a while, all the available places will be used the process automatically reuses old locations. This way, the database will not grow in size and therefore requires no maintenance. RRDtool works with with Round Robin Databases (RRDs). It stores and retrieve data from them.

NMIS uses RRDtool as the database for SNMP information and the graphing tool. Since NMIS does all of requests to and from RRDtool, the average NMIS user has little or no interaction with RRDtool.

Q. Does NMIS work on Microsoft Windows (any version)?

A. You aren't going to like this. The short answer is no. The long answer is yes, but.....

NMIS uses the Perl language and an Open Source Perl distribution is available from ActiveState. However NMIS uses a couple of modules to provide Time to the program flow and these do so at very high resolutio From what we (those good people on the mailing list) can tell it seems that these modules are Unix/Linux specific. Its seems that everyone uses a Unix/Linux variant at this time (usually Red Hat or Debian althoug Keith wrote NMIS on Solaris i386) so no-one has gotten around to fixing the problem. Some people have reported problems with RRDtool on Microsoft platforms also but this is unconfirmed.

A couple of other points. *One*.The general consensus is that this program is well suited to Linux/Unix environment and you are broadly encouraged to do so as well. *Two*. Since very few people, or perhaps no-one, is using Microsoft Windows then support is going to be a bit limited. *Three*. It would be great if you can get it working on Microsoft Windows and let us know. Then we can get a more people using NMIS and develop the software even further than it is now.

Q. Why Don't ping and traceroute work?

NMIS was written on Solaris so until version 2.2 is finalised there are some hard-coded file locations you might have to change in /bin/admin.pl change the file locations for ping, traceroute etc. to suit your machine (the command 'which ping' will tell you where it lives)

You may need to change the syntax of ping - for eample the following works in Linux at about line 89 of /cgi-bin/admin.pl

```
if ($admin eq "ping") {
                $adminoutput=`/bin/ping -c 5 -s 64 $node`;
        }
```

The same applies for other commands. This will be automagically fixed in 2.2 :)

Q. Why Aren't I getting any stored reports?

Change the first line of run-reports.sh to #!/bin/sh if you don't have ksh at /bin/ksh ('which ksh' will tell you where it is or report an error if you don't have it at all.) Check that the path listed in the script will find all the commands listed in the file eg. date, col etc. Again, you can find their locations with 'which *commandname*

Q. How is the Group Summary metric derived ?

Reachability is 40%, health is 40% and availability is 20%.

Q. What are the weighting for the Health Metric ?

A. This more fully explained in the NMIS Changelog but the basic info is:

reachability is 30%
interface utilisation is 20%
response time is 20%
availability is 10%
CPU is 10%
MEM is 10%

Interface Utilisation consists of all interface input and output utilisation are each subtracted from 100, added together and then averaged, this number is the interface weight, this will contribute 20% of the health metric and should be quite responsive to interface utilisation.

Q. Can NMIS handle SNMP traps and syslog?

A. No. But the NMIS event subsystem could be used in a real time event system. It would be possible to integrate NMIS into the Perl Syslog Daemon @ http://www.weirdness.net/acd/code/syslogacd or into NET-SNMP which includes a trap daemon, some details are available @ http://www.sins.com.au/nmis/net-snmp.html.

NMIS - Features

Last updated 21 June 2001	Online Version	NMIS Home Page

The following is a list of NMIS features. This is by no means comprehensive but provides and idea of wha NMIS can do.

General

- The entire network is summarised into a single metric, which indicates reachability, availability and health of all network devices being managed by NMIS.
- Summary page for entire network with reachability, availability, health, response time metrics.
- Summary pages of devices including device information, health graph, and interface summary.
- Can be distributed across multiple "polling servers" by using included programs.
- Policy based event and escalation.

Performance and Fault

- Integrated Fault and Performance Management.
- Color coded events, status for at a glance interpretation.
- Graphing of Interface, CPU, Memory stats for Cisco Routers and Switches.
- Graphs can be drilled into.
- Graphs produced on the fly.
- Graphs can have varying lengths from 2hours to 1 year.
- Interface statistics are returned in Utilisation and/or bits per second.
- Response time graphed and metrics for health and availability generated from statistics collected.
- Threshold engine which send alerts on certain thresholds.
- Escalation subsystem based on device groups which provides a great deal of granularity.
- Varying event levels for different device types.
- Alert events are issued for device down or interface down.
- Event levels are set according to how important the device is.
- Events are "State full" including thresholds, meaning that an event is only issued once.
- Notification engine can be expanded to handle any "command line" notification method, including email, paging, signs, speakers, etc.
- Integrated logging facility to view NMIS events and syslog messages.
- A list of current events is available and there is an escalation level and time the event has been acti
- Event logging
- Outage time calculated for each down event
- Planned outages can be put in so alerts are not issued

Configuration

- Find function which searches interface information for node name, interface name, description, type IP address, for matching interfaces.
- Interface information includes IP address information.
- Dynamic handling of ifIndex changes and difficult SNMP interface handling
- Checking of changes to device details.
- NMIS stores contacts and location information which links to the SNMP sysContact and sysLocatior MIBS.
- Produces DNS and Host records from the collected IP addressing information
- Produces DNS LOC records for "visible" traceroute utilities.

Reporting

- Reports for utilisation, outages, etc
- Snapshot and dynamic reporting for metrics on all devices and groups of devices.

NMIS - Change Log

Last updated 21 June 2001	Online Version	NMIS Home Page

Changes

- LOTS of tidy up in general.
- Stopped loading MIBS all the time.
- NMIS support multiple config files ALOT better, it was basically fine for nmis.pl but not the CGI stuff, should be fine now.
- Simplified the algorithm calculating group status. This is now done by weighting each status level and adding them together and then averaging. You can tweak the results in the overallNodeStatus subroutine.
- Fixed up reports.pl and run-reports.sh, I would actually like to rewrite the entire reporting subsystem, but it won't happen in a hurry.
- Deprecated some config elements and subroutines which were redundant.
- Escalation now done based on Group instead of sysLocation, more logical! Maybe notification should too!
- Converted links to CSV! Careful with conversion, should be fairly easy.
- Fixed links functionality!
- Added checkConfig subroutine and type=config nmis directive which checks the nmis configuration file and makes sure everything is OK. This will be run every time type=update is run but passively.
- Added printApache subroutine and type=apache nmis directive which displays a sample Apache configuration for the NMIS configuration file.
- Added metrics.pl which is a tiny dash with just the basic NMIS network metrics.
- Multilevel debug now, debug is 0-9 where 0 is false, and then the level of debugging you would like. This means more debugging can be added at varying levels of verbosity.
- Lots of code tidy ups, indenting, whitespace, etc to make it a little easier to follow.
- Zoomed up nmiscgi.pl a little by adding arrow icons for metrics, ie red is decreasing, green increasing. I have put this through all metrics, to help indicate where problems are coming from.
- Added config entries for icons and changed all icons in cgi scripts to reference config file.
- Changed getGroupSummary to handle statics which might be NaN, this was effecting color display for valid stats.
- Reweighted getGroupSummary total, so that reachability is 40%, health is 40% and availability is 20%. This should produce a slightly truer metric.
- ifTypes are now handled as a CSV file, this had removed the need for loadInterfaceTypes in NMIS.pl. It is necessary to ensure that iftypes.csv is copied to the conf directory and the ifTypes table entries are included in the nmis.conf, samples have been included in the nmis-sample.conf. This allows new interface types to be added without changing the code or compiling new MIBS.
- Updated whole bunches of nmiscgi.pl, updated the style and revamped it a little bit, looks a bit better now.

- Now using "auto directory finding option" so don't have to edit scripts as much on install, uses Perl FindBin.
- Many new config.dat options and configuration items exist and now called nmis.conf and in <nmis base>/conf
- Added a Generic event type to handle any new UP/Down events you might like, also created event.pl, a CGI script for distributed event management, or creating events in NMIS for a device from other processes, like Unix daemon down/up. Proactive events are already generic ie any event starting with "Proactive" passed with a level=whatever will create an event, and passed with level=normal will close it. Could be handy.
- Simple SNMP error checking in updateUptime which creates an SNMP Down event and checks for SNMP Down if SNMP is up. Prevents problems with people changing the SNMP community string or disabling SNMP. Assumes that if SNMP is up for UpTime ie working for system SNMP variables then it should be working for everything.
- NMIS now has Web based configuration using view.pl. View.pl allows editing of the CSV configuration files. Will hopefully add abilitiy to edit the nmis.conf file too.
- Health metric now includes interface component, all interface input and output utilisation are each subtracted from 100, added together and then averaged, this number is the interface weight, this will contribute 20% of the health metric and should be quite responsive to interface utilisation. I also think this will assist in pinpointing link utilisation problems through the use of the health metric.
- Health metrics have been re-weighted to better suit pinpointing network problems with the health metric. Weighted as follows:
    - reachability is 30%
    - interface utilisation is 20%
    - response time is 20%
    - availability is 10%
    - CPU is 10%
    - MEM is 10%
- Updated logs.pl with a better parsing algorithm, handles the weird messages in syslog which routers and switches send.
- Added map.pl which enables NMIS to display group status colored icons for each group on a HTML map using CSS. Icons will change color with group status, also display summary network metrics and allows drill in to NMIS for each group. (not auto-discovery or drag and drop). Adds a bunch of config.dat elements and requires setting up map.csv with icon placement co-ordinates.
- Changed the way events are handled by implementing eventPolicy subsystem, all event levels and notifications are now controled through the events.csv file.
- Added real escalation subsystem with escalationPolicy and runEscalate, modularised stuff a bit more to support this.
- Optimised the runThreshold subroutine a little bit.
- Added in-addr.arpa and DNS LOC entries to the DNS function in nmiscgi.pl.
- Error checking added to csv.pm to validate data in each record.
- logs.pl fixed up to do handle unknown records a little better parse.
- nmiscgi.pl now displays summary stats for each group and a raising falling indicator for metric (could do with some graphics I am sure but more to do).

- Improved support for the Cisco Catalyst IOS switches, not really tested.
- NMIS Large Dash now has a config option.
- NMIS and nmiscgi.pl now uses embedded PNG graphics generated at runtime, this should improve polling scaling. This also assists in distributed management, etc. All automatic graphing has been turned off in nmis.pl by setting the $graph = "false". This just improved the poll cycle from real 0m10.33s, user 0m6.72s, sys 0m0.43s to real 0m5.61s, user 0m2.31s, sys 0m0.32s.
- Automatically handling of interfaces which have been shutdown between update cycles, this will recreate the interface files on demand.
- Added query.pl to enable NMIS to be used in a distributed manner, returns easily passed text data over HTTP. Eg query.pl?query=list or query.pl?query=status&group=Sydney.
- Added summary.pl to enable NMIS to be accessed by ANY HTML device like WAP! Just provides basic lists and summary information.
- query.pl and summary.pl show how easy it is to develop mini NMS apps which use the NMIS "api" to easily access all the collected information.
- Added email field to contacts table.
- Added support for default locations and contacts for devices without sysLocation and sysContact being set.
- notify subroutine now uses the email field from the contacts table to issue notifications.
- Added CGI graphing to nmiscgi.pl which does the NMIS graphing on the fly, cool, now you can embed NMIS graphs where ever you want! Have started embedding this into all CGI scripts which display graphs. All NMIS graphs will be drawn on the fly.
- Changed the loadConfiguration subroutine to handle variables in the configuration file. This will make installation a little easier for people.
- Added Interface thresholding for input utilisation, output utilisation and interface availability.
- Updated the nmiscgi.pl with a summarised dash and left the nmisMenuLarge as another option. Also created a new summary for each group, allows a little more the concept of drill in!
- Enhanced NMIS documentation.
- Brian gave NMIS a logo.
- Tested with RRDTool 1.0.33.
- Added IP addresses to interface info and fixed up the "find" function so that you can search the interfaces for an IP address (this could be handy for big networks).
- New PING routine using Perl Ping with ms timeout too! Code provided by Richard Kuehnle (this requires root privilages for the NMIS programs)
- Added support for a enterprises file for identifying vendors.
- Worked on createInterfaceFile and added ability to capture IP addresses from the ipAddrTable MIB.
- Added a view table function which allows the easy integration of other data sources.
- Added Locations and Contacts Tables which use the view table script.
- Did a fairbit of Cascading Style Sheet Integration into nmiscgi.pl. This means if you want to change the way NMIS looks you can just edit the NMIS Style Sheet nmis.css and wallah.
- Fixed up that a node with collect = false would not store response time stats.

- Fixed up SNMPv2 Support, now getNodeInfo tests for SNMPv2 support and then if available caches as part of system attributes.
- Added more MIBs to the OID files, and enhanced the existing MIBS to include other common interesting things.
- Fixed it so that the nodeType, netType, and role from the nodes file overrides the getNodeInfo information.
- NMIS is considerable modularised. This is to make the system a little easier to improve with new changes only being added in one file not effecting other files. This includes new packages:
    - ip.pm
    - func.pm
    - csv.pm
    - ping.pm
    - web.pm (replacing htmllib.pm)
- Added an IP subnetting program to assist with those nasty subnetting calculations.
- Memory stats now contribute to health statistics.
- Added a notification method for paging. There is now a sendPagerMessage to integrate into an existing paging system.
- SNMP_Simple now included in the NMIS distribution library directories (much easier).
- This includes several minor fixes to the logic and some more considerable changes like using SNMPv2 for collection on Routers and Switches. New options in the config file.
- NMIS now produces a single metric which summarises the network.
- The first version of the documentation has been added.
- FAQ's added.
- The PING routine seems to be the biggest problem and I am working on a better cross platform fix but for now Linux users will have to apply there own patch as there seems to be some vast differences to PING across Solaris and Linux.
- NMIS has been tested and updated to work with RRDTool 1.0.27, I found some unexplained core dumps with 1.0.28 so have decided to go back one and investigate those further.
- Thanks to the Contributors for assisting in the debugging and development of NMIS.

NMIS Apache Configuration

Last updated 21 June 2001	Online Version	NMIS Home Page

Below is a sample configuration for Apache to implement IP address and user name security. Its pretty straight forward. To produce this configuration you can run NMIS like nmis.pl type=apache which will produce a sample config with the nmis.conf or whichever file it is.

- Apache Docs
- Apache Notes
- NMIS Aliases
- NMIS Location Setup

Apache Docs

For more information on the listed Apache features read:

- **Apache Docs**
- Alias directive
- ScriptAlias directive
- Order directive
- Allow directive
- Deny directive
- AuthType directive
- AuthName directive
- AuthUserFile directive
- Require directive

Apache Notes

Usual Apache Config File!

```
<apache_root>/conf/httpd.conf

add a password to the users.dat file!

<apache_root>/bin/htpasswd /usr/local/nmis/conf/users.dat nmis

restart the daemon!

<apache_root>/bin/apachectl restart
```

NOTE:
<apache_root> is normally /usr/local/apache
the "bin" directory might be "sbin"
the "conf" directory might be "etc"
the httpd.conf might be split across httpd.conf, access.conf and srm.conf

NMIS Aliases

Alias /nmis/ "/usr/local/nmis/htdocs/"
ScriptAlias /cgi-nmis/ "/usr/local/nmis/cgi-bin/"

NMIS Location Setup

```
<Location "/cgi-nmis/nmiscgi.pl">
        ## For IP address based permissions
        Order deny,allow
        deny from all
        allow from 10.0.0.0/8 172.16.0.0/16 192.168.1.1 .sins.com.au
        ## For Username based authentication
        AuthType Basic
        AuthName "NMIS"
        AuthUserFile /usr/local/nmis/conf/users.dat
        Require valid-user
</Location>

<Location "/cgi-nmis/logs.pl">
        ## For IP address based permissions
        Order deny,allow
        deny from all
        allow from 10.0.0.0/8 172.16.0.0/16 192.168.1.1 .sins.com.au
        ## For Username based authentication
        AuthType Basic
        AuthName "NMIS"
        AuthUserFile /usr/local/nmis/conf/users.dat
        Require valid-user
</Location>

<Location "/cgi-nmis/admin.pl">
        ## For IP address based permissions
        Order deny,allow
        deny from all
        allow from 10.0.0.0/8 172.16.0.0/16 192.168.1.1 .sins.com.au
        ## For Username based authentication
        AuthType Basic
        AuthName "NMIS"
        AuthUserFile /usr/local/nmis/conf/users.dat
        Require valid-user
</Location>

<Location "/cgi-nmis/view.pl">
        ## For IP address based permissions
        Order deny,allow
        deny from all
        allow from 10.0.0.0/8 172.16.0.0/16 192.168.1.1 .sins.com.au
        ## For Username based authentication
        AuthType Basic
        AuthName "NMIS"
        AuthUserFile /usr/local/nmis/conf/users.dat
        Require valid-user
</Location>
```

NMIS Documentation

Last updated 21 June 2001	Online Version	NMIS Home Page

- NMIS Home Page
- Introduction
- Concepts
- Roles and Groups
- Health
- Events
- Thresholds
- Updates
- Interfaces

Introduction

NMIS stands for Network Management Information System. It is a Network Management System which performs multiple functions from the OSI Network Management Functional Areas, those being, Performance, Configuration, Fault. A primary function of NMIS is to make information about your network available quickly and instantly. Some of this network is provided "raw" other information is provided in a related manner.

It started as a SNMP polling and statistics viewer front-end to Tobi Oetiker's RRDTool. RRDTool replaces MRTG but doesn't include a front end and backend to handle SNMP polling and display resulting web pages etc. The original NMIS evolved quite rapidly to meet demands of production environments.

The backend, polling engine, uses SNMP to collect interface and health statistics for Cisco Routers, certain Cisco Catalyst Switches and Generic SNMP devices every 5 minutes. The collected statistics are stored in RRD's (Round Robin Databases) and ensures that devices are up, issues alerts, etc. The front end accesses the information stored in the RRD's and displays statistics the resulting graphs, reports, etc.

Both the front and back ends are highly extensible and features are easy to add as the structure is learnt. For example the backend was just collecting interface statistics every poll cycle, it was easy to add collection of health (cpu, memory, buffer, etc) and response time, availability.

NMIS uses a backend to collect data and maintain the information about the data. It relies on RRD for databases, additional tables are text based configuration information. The frontend is independant, it just reads information from the RRD's and text tables and displays the information. Simple.

It is intended that NMIS be low maintenance once it is running, it should just go and go. More work needs to be done on this but I think it going well so far.

Concepts

The basic concept is that NMIS collects interface, CPU, Memory, buffer and packet statistics from Cisco Routers and Switches, it is also capable of supporting generic SNMP MIB 2 collection. Getting slightly deeper, NMIS pings a device every poll cycle verifies that it is "up", this is called "reachability", it holds this in memory.

If no system information is available for the device it must be a new device so perform a capabilities discovery on the device, this is the subroutine getNodeInfo. Otherwise load the cached system information with the loadSystemFile then run the updateUptime subroutine which gets sysObjectID, sysUpTime and ifNumber, NMIS compares this with the cached information to see if the same number of interfaces are present, that the uptime has increased and that the sysObjectID is the same.

If the number of interfaces has changed run the createInterfaceFile subroutine to update this information. (This should send an configuration change event.)

If the sysObjectID has changed run the getNodeInfo subroutine. (This should send an configuration change event.)

If the sysUptime is less then the cached information sysObjectID has changed run the getNodeInfo subroutine. (This should send an node reload event.)

The runHealth subroutine is run, this collects CPU, Memory, buffers, etc, whatever is deemed necessary for that device type and stick it all in an RRD.

Then the runInterfaces subroutine is run, it loads the cached interface information, if none exists it creates it with createInterfaceFile. Then for each interface it collects ifDescr, ifOperStatus, ifInOctets and ifOutOctets. If the ifDescr is different, the cached interface information must be out of date (this is how shifting ifIndex is handled) create it again with createInterfaceFile. If the ifOperStatus shows down

when the interface is supposed to be up, raise an event. Otherwise store ifOperStatus, ifInOctets and ifOutOctets in an RRD, adding ifOperStatus to the total interface availability of the device.

After the interfaces are complete, calculate the response time for the device with another ping and store some health metrics in another RRD, we store the reachability of the device, the interface availability of the device, the responsetime and create a health metric from a simple algorithm which weights various collections and makes up a metric to indicate the overall health of that device, more on this in the health section.

Roles and Groups

The ability exists to put nodes into two types of groups, the first group is a role which is core, distribution and access, the second group is used to group devices together for reports, and general information. It is logical hat the second group be something like the building name or city/suburb of the device as this helps identify problem areas.

Roles play an important part in NMIS, they allow things to be weighted for events and various other functions. The concept of weighting according to role is simple, if it is a core device then it is important and should be treated as such, if it an access device then it is less important. The idea is to try and remove the noise, ie all events coming in at critical and which ones really are.

Health

The following statistics are considered part of the health of the device:

- Reachability - is it up or not;
- Availability - interface availability of all interface which are supposed to be up;
- Response Time;
- CPU;
- Memory;

All of these metrics are weighted and a health metric is created. This metric when compared over time should always indicate the relative health of the device. Interfaces which aren't being used should be shutdown so that the health metric remains realistic. The exact calculations can be seen in the runReachability subroutine.

Events

- Escalation
- Events based on device role
- Stateful

Thresholds

The thresholds routine runs whenever you like, it process the collected statistics in the RRDs and compares the numbers to stored thresholds and if exceeded raises an event for that device. The thresholds use the device role to weight the events.

Updates

Updates ensures that all the cached system and interface information is kept up to date. If the network is constantly changing then it should be run frequently, otherwise it could be run less frequently.

Interfaces

Interfaces which aren't in use should be shutdown (admin down) so that NMIS doesn't think it is supposed to manage them. A simple lookup is done on interface types to determine if NMIS should collect statistics on them. This is done during the createInterfaceFile subroutine.

NMIS Help

Last updated 21 June 2001	Online Version	NMIS Home Page

General:

- Menu
- Summary
- Health
- Find
- Current Events
- Event Log
- Reports
- Outages
- Links
- Logs
- NMIS Logs

Tables:

- Nodes
- Links
- Locations
- Contacts
- Event Policy
- Escalation Policy

Extras:

- Draw Graph
- IP
- metrics.pl
- map.pl
- view.pl
- summary.pl
- event.pl
- query.pl

Menu

From the menu you can access the features and functionality of NMIS, there are buttons and pulldowns which allow various ways to access NMIS items. Many things in NMIS are clickable allowing you to drill in and around looking for information and answers.

One of NMIS key goals is to summarise the network into a single metric. This has been achieved and allows network managers and engineers to watch this metric for problems in the network as the metric changes. These metrics are shown on the NMIS Dashboard. You can view a small dash or a large dash, if you prefer the large dash by default change the nmis.conf config item "show_large_menu" to "true".

Summary

This displays summary details about a node. This includes some configuration information, a health graph and an interface table. If you do not wish to see the uncollected interfaces set the nmis.conf config item "show_non_collected_interfaces" to "false".

Graphs are drillable.

Health

This displays more detailed graphs about the health of a device. If available, CPU, Memory, packets switched or routed, etc.

Find

Allows you to search the NMIS data for all interfaces with the keywords in IP address, description, interface name, subnet mask. This is a simple way to see exactly how many Gigabit interfaces you have and the IP addresses for each.

This becomes quite a powerful tool by being able to identify which routers have which IP address and IP subnet.

Current Events

Shows a list of active events in the NMIS event state table. This is what is wrong with the network right now. You can acknowledge and un-acknowledge events to stop them escalating. Events can't be deleted, this is quite deliberate, without them in the log they will keep occurring.

Event Log

A list of previous events color coded and drillable. Handy!

Reports

Produces dynamic reports and lists the snapshot reports if it has been enabled (done through the reports.pl and run-reports.sh scripts).

Outages

Add planned outages to NMIS so NMIS won't decrease your network metrics.

Links

When configured lists a set of links on NMIS and show summary information, another nice way to view the network information.

Logs

Allows you to view color coded drillable logs of Cisco Syslog information (if configured) and other logs you might like to have on your web browser.

NMIS Log

Allows you to view NMIS log information for errors, etc. A great place to look if things are going wrong.

Nodes

You can add, edit and delete entries to the NMIS Nodes table.

Links

You can add, edit and delete entries to the NMIS Links table.

Locations

You can add, edit and delete entries to the NMIS Locations table. Allows information to be stored about locations for the network. Links to sysLocation. Lets map that logical to physical.

Contacts

You can add, edit and delete entries to the NMIS Contacts table. Allows information to be stored about contacts for the network. Links to sysContact.

Event Policy

You can add, edit and delete entries to the NMIS Event Policy table. This controls how events are processed by NMIS, what event level each node type and role type will generate and weather or not it will send alerts.

Escalation Policy

You can add, edit and delete entries to the NMIS Escalation Policy table. This controls how events are escalated by NMIS, what each node type and role type will generate and weather or not and how it will escalate unacknowledged events.

IP

The IP tool allows you to view and calculate IP addressing information for IP subnetting. Handy if you are wondering what the broadcast address is for IP address 10.24.51.193 mask 255.255.255.224

Draw Graph

Draw graph is a subroutine used by NMIS to produce graphs, you can embed this in any HTML you like and can be used to allow distributed access, just to a view HTML source on one of the web pages to see how. Basically you call it like this:

```
<img border="0" alt="Device Health"
src="http://nmis_server/cgi-nmis/nmiscgi.pl?file=nmis.conf&type=drawgraph&
node=c1000&graph=health&length=2days&start=0&end=0&width=500&height=100">
```

Where:

- file is nmis config file defaults to nmis.conf
- type must be "drawgraph"
- node is the node name
- graph can be health, cpu, memory, interface, etc.
- length is AT style, ie 2days, 1day, 2hours, etc.
- start and end default to 0 but you can start x seconds ago and end y seconds ago.
- height and width are the requested size of the image, which will actually be the size of the graph not the resulting image.

metrics.pl

This NMIS CGI script displays a simple metric table, handy for keeping visible somewhere like an active desktop.

map.pl

This displays a map and icons of each configured group. You edit the map.csv to control this. You should create and entry for each Group and find a suitable Map background icon. The colors represent the status, etc.

This is intended to put up on a screen for "at a glance" checking on the network. Handy for a basic NOC monitoring function, ie colored icons change to represent the problem and you can drill in.

view.pl

This allow NMIS to view and edit the CSV files used to control NMIS.

summary.pl

This produces a VERY basic summary of the NMIS for group and node status, alows drill around, intended for WAP, PDA Web Browsers, dialup, etc.

event.pl

This allows integration for other systems to send events to NMIS over HTTP, by calling URL's with required information. Quite handy to allow NMIS event and escalation subsystem to be used for this! Ie monitoring something else, oh its down, I will tell NMIS to add it to its state table and check the policy and send an alert if required.

query.pl

This can be used to allow NMIS to be used in a distributed manner, this CGI can be called from remote machines to determine what this NMIS is monitoring and what the status and health of devices are.

NMIS To Do

Last updated 25 February 2002	Online Version

To Do

- Write an installation script.
- Make polling engine multi-threaded.
- Enhance documentation.
- Use NET-SNMP as the SNMP Package
- Time and Date on Dash Board

Health

Make health include metrics for interface utilisation, could be done by doing utilisation summarystats for ea weighted out of 100 where 50 utilised would equal 50, etc, which then contributed to a total number divided the total health metric.

Locations

Add timezone for each location, in the form of UTC or Local name.

Complete Locations lat and lon from Encarta

Health

Every 5 minutes calculate the health metric for each group & whole network and stick in an RRD allowing g health on the dash board.

Collection Policy

Create a CSV with interface types and node types and some flags for defining the collection policy for devi

ifType	device_type	role
default	any	any
ethernet	switch	any
ethernet	switch	any
atm	any	any

summarystats

I would actually like to rewrite the summary stats routine to return a "summarystats hash".

Interface Stats

For each interface which supports it collect in/out frames, broadcast, unicast. This would allow determinati would also allow thresholds to be set for percentage of broadcast traffic on network.

More Notes:

An excellent idea, this would be very good information, this is already
on the todo list but there are few "dependancies":

1. The poller should be enhanced first to be multi-threaded which will
enable improved performance for SNMP and increase the number of mibs
polled per second per node. I think this is the next major improvement
required. That along with making NMIS a daemon which go hand in hand.

2. I would also like to support packet stats, ie pkt in and out, as well
as unicast and non unicast. There are different error stats per
interface type.

3. Not all interfaces support full if-mib stats, ie frame relay
subinterfaces on Cisco devices only support ifInOctets and ifOutOctets,
all other stats are in the Frame Relay mibs. This is also true for
other interface types like ATM. So the solution is to keep the extended
stats in a different RRD for each interface, I would envisage that there
would be the current interface, extra bit (pkts, errors, etc), specific
(frame-relay, ATM, etherlike, etc), so one interface may have 1
interface RRD while another my have 3. This could be implemented in
phases too I suppose.

4. Yes putting error rates in the node health metric would be excellent.

5. Other metrics could be calculated for nodes and the network:
* like per interface/link average packet size;
* entire network average packet size;
* error rates per interface/network (this equates to the much heard
never seen "accuracy" metric);
* lots of others that we might think of later.

6. Thresholding on errors should be done (easy).

Create RRD

I had been thinking about making this config options and putting the formulas in the code.

NMIS Command Line

simple script to provide status of nodes and basic info like summary.pl but commandline oriented.

Thresholds

Put the threshold code seperate for just using in nmis.pl

HOST MIB

Add support for HOST-MIB, then get CPU and MEM working dynamically, get DISK mapping working for c

Add a control file like the interface.dat which maps CPU and MEM and DISK, also tracks inventory change

```
Eric,

Yep, this was used pre Event Policy table, which is when I figured the event
policy should be used.  I think this is only used by nmiscgi.pl and looks
redundant.  If an event exist for a node, then just get the level from the
event state table.  Should be a small patch.


Regards


Keith
-----Original Message-----
From: imlnetnz [mailto:imlnetnz@yahoo.co.nz]
Sent: 18 December 2001 18:15
To: nmis_users@yahoogroups.com
Subject: [nmis_users] eventlevel


Keith - I am not sure why the subroutine eventlevel in NMIS.pm
appears to overule the event levels as defined in the policy table -
is there  a good reason for this, have I misread the code, or was
this something to get the project up and running out of the box ?

regards

Eric
```

Interface Speeds and SNMP Spikes

Yes and No, best handled in RRD but not so easy, sort of need to handle in NMIS and RRD, ie if NMIS det checks last RRD value and sets current interface poll to 0 which should reset the RRD and remove the spil

Nasty little SNMP! I think the problem is when the SNMP counter gets reset and RRD doesn't know how t it has a small spike as it thinks the counter has wrapped.

Limiting the CDEF to the speed of the link is a good idea, but for some interface types it is not accurate, ie capability. I will add this to the TODO list and ponder it a little more, will have to put some code in to handl

```
Greetings,

      I have experienced odd behavior in graphs if a link/router goes down
unexpectedly. I'm not sure exactly why, but when you view the Bits graph
there will be a large spike (larger than the link is capable of
providing). I don't believe it is an issue with NMIS itself, I believe
the router is reporting incorrect figures... but I digress.

      If anyone would like it, I have changed a few CDEF's in nmiscgi.pl to
not allow throughput greater throughput figures than the link is capable
of as defined by the interfaceTable.

      In nmiscgi.pl, just comment the current input/outputBits CDEF's under
the drawrrd sub routine and add the two conditional ones below (I'd
offer up a patch, but there a a few other changes we've already made
here):

#"CDEF:inputBits=input,8,*",
#"CDEF:outputBits=output,8,*",
"CDEF:inputBitsTmp=input,8,*",
"CDEF:inputBits=inputBitsTmp,$NMIS::interfaceTable{$tmpifDescr}{ifSpeed},GT,UNKN,inputBitsTmp,
"CDEF:outputBitsTmp=output,8,*",
"CDEF:outputBits=outputBitsTmp,$NMIS::interfaceTable{$tmpifDescr}{ifSpeed},GT,UNKN,outputBitsT


Now, you will still see spikes with these changes, but they will not
exceed the speed of the link (i.e. laws of physics still prevail).

Thanks to Keith and all who have worked on this project - it is an
irreplaceable tool! If I'm the only one who has seen these spikes simply
disregard this message as the ramblings of an idiot.

regards,
david
```

Guilherme's Patch

I had the same behavior, your fix seemed to make it work right.
BTW I made some other hacks on the nmis I like to share:

1-Multithread collection and update - I believe it was on the TODO list.
To use it just add the option "mthread=true" on the nmis.pl command line for
the types update and collect.
It will make nmis.pl to fork one instance for each node.
It should be warned that it is resource consuming, I had to increase my host
memory from 64MB to 192MB to avoid swapping. But I achieved a performance gain
from 6 to 1 minute to collect from almost hundred routers

2-Fix for 3Com routers wich describe virtual interfaces with the same string as
the actual physical interface. Because of this, all frame relay (I think X.25,
ATM and ISDN also) interfaces were not correctly collected or displayed. Now
they are hard-coded to be interpreted as virtual interfaces.

3-Fix for groups and devices summary statistics when some item have "nan"
value. The fix were also applied to some reports. I believe that almost all
cases are treated correctly

5-Display the IP addresses together with devices in large dash. And, for the
large dash the telnet url uses the IP address, rather than host name (it avoid
problems in networks that don´t use DNS records for routers and switches). I am
working on using this feature on the other screens and reports.

6-The options to change files and tables are hidden in nmiscgi.pl. I didn´t
anyone messing around with them without root access to server. I know that by
default there are no permissions for this to happen, but it made the dash
screens lighter.

Attached is a diff file. I am sorry to not have one diff for each feature.
I want to hear your impressions.

regards,
Guilherme Chehab

View Sorting

```
--- view.pl    2001/06/23 01:35:28     1.1
+++ view.pl     2001/10/11 15:21:22     1.2
@@ -75,6 +75,31 @@

 exit;

+sub alphanumerically {
+       local($&, $`, $', $1, $2, $3, $4);
+       # Sort numbers numerically
+       return $a cmp $b if $a !~ /\D/ && $b !~ /\D/;
+       # Sort IP addresses numerically within each dotted quad
+       if ($a =~ /^(\d+)\.(\d+)\.(\d+)\.(\d+)$/) {
+               my($a1, $a2, $a3, $a4) = ($1, $2, $3, $4);
+               if ($b =~ /^(\d+)\.(\d+)\.(\d+)\.(\d+)$/) {
+                       my($b1, $b2, $b3, $b4) = ($1, $2, $3, $4);
+                       return ($a1 <=> $b1) && ($a2 <=> $b2)
+                               && ($a3 <=> $b3) && ($a4 <=> $b4);
+               }
+       }
+       # Handle things like Level1, ..., Level10
+       if ($a =~ /^(.*\D)(\d+)$/) {
+               my($a1, $a2) = ($1, $2);
+               if ($b =~ /^(.*\D)(\d+)$/) {
+                       my($b1, $b2) = ($1, $2);
+                       return $a2 <=> $b2 if $a1 eq $b1;
+               }
+       }
+       # Default is to sort alphabetically
+       return $a cmp $b;
+}
+
 sub editRow {
        my %args = @_;
        my $table = $args{table};
@@ -95,7 +120,7 @@

        if ( $edit ne "delete" ) {
                if ( $row eq "add" and $edit eq "true" ) {
-                       foreach $key (sort(keys %table_data)) {
+                       foreach $key (sort alphanumerically (keys %table_data)) {
                                if ( not defined $tmp ) { $tmp = $table_data{$key}; }
                        }
                }
@@ -103,7 +128,7 @@
                        $tmp = $table_data{$row};
                }
                # need to get a row to build the menu.
-               foreach $field (sort(keys %$tmp)) {
+               foreach $field (sort alphanumerically (keys %$tmp)) {
                        $table_data{$row}{$field} = $FORM{$field};
                        #print STDERR returnTime." editRow, field=$field row=$row table=$table
                }
@@ -168,7 +193,7 @@
                }
        }
        $i = 0;
-       foreach $head (sort (keys %$tmp_key)) {
+       foreach $head (sort alphanumerically (keys %$tmp_key)) {
                $headers[$i] = $head;
                ++$i;
        }
@@ -250,7 +275,7 @@
        cssTableStart("white");

        #Display each data Row
-       foreach $key ( sort (keys %table_data) ) {
+       foreach $key ( sort alphanumerically (keys %table_data) ) {
                print "\n";
                ++$counter;
                ++$pass;
@@ -267,7 +292,7 @@
                        }

                        $i = $c;
```

DONE

SENDMAIL

Perl based sendmail for better mail control.

DONE

Enhanced SNMPv2c

Add support for the HC MIBS, in fact test and see if HC mibs are supported in all SNMPv2c devices.

```
mibdump.pl and generating the new OID files, see the FAQ,
change
            SNMP_MIB::loadmib($argue{mibdir}, "IF-MIB-V1SMI.my");
for
            SNMP_MIB::loadmib($argue{mibdir}, "IF-MIB.txt");

subroutine runInterface
line 548
                  if ( $NMIS::systemTable{snmpVer} eq "SNMPv2" ) {
                        # do the SNMP stuffy to get the standard stats
                        (      $ifStats{ifDescr},
                              $ifStats{ifOperStatus},
                              $ifStats{ifAdminStatus},
                              $ifStats{ifInOctets},
                              $ifStats{ifOutOctets}
                        ) = $session->snmpget(
                              'ifDescr'.".$interfaceTable{$interface}{ifIndex}",
                              'ifOperStatus'.".$interfaceTable{$interface}{ifIndex}",
                              'ifAdminStatus'.".$interfaceTable{$interface}{ifIndex}",
                              'ifHCInOctets'.".$interfaceTable{$interface}{ifIndex}",
                              'ifHCOutOctets'.".$interfaceTable{$interface}{ifIndex}"
                        );
                  }
                  else {
                        # do the SNMP stuffy to get the standard stats
                        (      $ifStats{ifDescr},
                              $ifStats{ifOperStatus},
                              $ifStats{ifAdminStatus},
                              $ifStats{ifInOctets},
                              $ifStats{ifOutOctets}
                        ) = $session->snmpget(
                              'ifDescr'.".$interfaceTable{$interface}{ifIndex}",
                              'ifOperStatus'.".$interfaceTable{$interface}{ifIndex}",
                              'ifAdminStatus'.".$interfaceTable{$interface}{ifIndex}",
                              'ifInOctets'.".$interfaceTable{$interface}{ifIndex}",
                              'ifOutOctets'.".$interfaceTable{$interface}{ifIndex}"
                        );
                  }
```

DONE

CSS Changes

CSS is now case sensitive, ie normal is not Normal! Have to change CSS to suit. Problem with IE6.0

DONE

Hardcoded script name

Yep, that should be $ENV{SCRIPT_NAME} instead of /cgi-nmis/etc...

Keith

-----Original Message-----
From: Kuehnle, Richard [mailto:rkuehnle@us.tiauto.com]
Sent: 03 October 2001 18:49
To: 'nmis_users@ahoogroups.com'
Subject: [nmis_users] configuration problem is nmiscgi.pl

```
Keith,

I am sure that you have seen this one already.  I only noticed it because I am not using
default cgi path, as you see, I am using /cgi-nmis-2/.

Rich Kuehnle
Network Manager
TI Group Automotive System, LLC.
810.755.8402

-----Original Message-----
From: root [mailto:root@tiasnadtcap91.na.tiautomotive.com]
Sent: Wednesday, October 03, 2001 9:44 PM
To: rkuehnle@us.tiauto.com
Subject:

*** nmiscgi.pl      Wed Oct  3 19:07:50 2001
--- nmiscgi.pl.bad      Wed Oct  3 21:43:13 2001
***************
*** 259,265 ****
        # If the heading isn't blank then there must be a graph type for it.
        if ( $heading ne "" ) {
              #KS 11 Mar 2001 NEW Embedded graphics, none of this dump in a temp directo
!             $graphLink="<img border=\"0\"
src=\"\/cgi-nmis-2\/nmiscgi.pl?type=drawgraph&node=$node&graph=$graphtype&le
ngth=$graphlength&start=$start_time&end=$end_time&width=660&height=150&inter
face=$interface\">";
        }
        else {
              $graphLink="Other graph types not yet supported\n";
--- 259,265 ----
        # If the heading isn't blank then there must be a graph type for it.
        if ( $heading ne "" ) {
              #KS 11 Mar 2001 NEW Embedded graphics, none of this dump in a temp directo
!             $graphLink="<img border=\"0\"
src=\"\/cgi-nmis\/nmiscgi.pl?type=drawgraph&node=$node&graph=$graphtype&leng
th=$graphlength&start=$start_time&end=$end_time&width=660&height=150&interfa
ce=$interface\">";
        }
        else {
              $graphLink="Other graph types not yet supported\n";
```

DONE With Model support and a new model called Catalyst5000Sup3

Catalyst Support

If you are using 6000 series switches and you click on health statistics
then try to drill into the cpu stats you will get a message:
Graph type not supported yet.

Line 1314 of nmiscgi.pl reads :

```
<a
href="$ENV{SCRIPT_NAME}?file=$conf&type=graph&graphtype=switch&graphlength=2
days&node=$node">
```

it should read :

```
<a
href="$ENV{SCRIPT_NAME}?file=$conf&type=graph&graphtype=cpu&graphlength=2day
s&node=$node">
```

Also previously several people have asked about Cat 5000 switches and what
snmp they support. We are now using some
Supervisor 3's in the cat 5000's and the mibs are identical to a Cat 6000.
(cpu, memory stats, traffic and topology)

We added this to nmis.pl to support them and NMIS trests them as a 6000 .
(There has to be a better way though.)

in sub getNodeInfo line :

```
      # Checking on the Model Type

elsif (       ( $NMIS::systemTable{sysDescr} =~ /WS-C5/i ) &&
                        ( $NMIS::systemTable{sysDescr} =~ /6.1/i )
            ){
                  $NMIS::systemTable{nodeModel} = "Catalyst6000";
                  $NMIS::systemTable{nodeType} = "switch";
                  $NMIS::systemTable{netType} = "lan";
                  $NMIS::systemTable{supported} = "true";
            }
```

DONE

runPing

```
sub runPing{
        my $node = shift;
        my $retries = 3;
        my $sleep = 15;
        my $i;
        if ($debug) { print returnTime." Starting Pinging with $retries retries.\n"; }
        $pingresult = 0;

        # do a ping $retries times.
        for ($i=1;$i<=$retries;++$i) {
                $pingresult = ping(node => $node, timeout => 5, debug => $debug );
                if ( $pingresult != 100 ) {
                        # Sleep a bit and try a second time.
                        if ($debug>1) { print "Sleeping $sleep seconds\n"; }
                        sleep $sleep;
                } else {
                        $i = $retries;
                }
        }

        if ( $pingresult != 100 ) {
                # Device is down
                $pingresult=0;
                if ($debug) { print returnTime." Pinging Failed $node $NMIS::systemTable{role
                notify(node => $node, role => $NMIS::systemTable{roleType}, type => $NMIS::sys
                # Device is down only update the runReachability if its an interface.
                if ( $type eq "interface" ) { &runReachability; }
        } else {
                # Device is UP!
                checkEvent(node => $node, role => $NMIS::systemTable{roleType}, type => $NMIS
        }
}
```

DONE

ping.pm

```
      my $r;
      my $num_tries = 5;
      $ping = Net::Ping->new("icmp", $timeout, $packetsize);
      while ($r<=$num_tries and $result != 100;) {
            if ( $debug eq "verbose" ) { print returnTime."
Pinging $node timeout $timeout\n"; }
            if ( $ping->ping($node, $timeout) ) { $result = 100; }
            else { $result = 0; }
            ++$r
      }
      $ping->close;
```

```
DONE
```

```
Keith - I am thinking that in sub thresholdResponse, NMIS should keep
the level at 'normal' or 'level=1' for response times, rather than
increment for the core and distribution node types ??

Code is..

      if ( $role eq "core" ) { $level + 2; }
      elsif ( $role eq "distribution" ) { $level + 1; }

I suggest maybe it should be this, same as all the other thresholds.

      if ( $level == 1 ) { $level = 1; }
      elsif ( $role eq "core" ) { $level = $level + 2; }
      elsif ( $role eq "distribution" ) { $level = $level + 1; }

Eric
```

```
DONE

Keith, IMHO a minor improvement would be to strip newlines from the
logmessage subroutine so that any (SNMP) error messages are formatted
appropriately in the log...

NMIS.pm

sub logMessage {
      my $string = shift;
      open(DATA, ">>$NMIS::config{nmis_log}") or warn returnTime."
logMessage, Couldn't not open log file $NMIS::config{nmis_log}. $!\n";
      for ($string) { s/\n+//g; }      #remove all embedded newlines
      print DATA &returnDateStamp.",$string\n";
      close(DATA);
} # end logMessage
```

DONE

```
I believe I've found a couple of instances of hard-coded url paths in
cgi-bin/nmiscgi.pl and cgi-bin/logs.pl.  Diff for nmiscgi.pl is:

262c262
<               $graphLink="<img border=\"0\" src=\"\/cgi-
nmis\/nmiscgi.pl?type=drawgraph&node=$node&graph=$graphtype&le
ngth=$graphlength&start=$start_time&end=$end_time&width=660&height=150
&interface=$interface\">";
---
>               $graphLink="<img border=\"0\" src=\"$NMIS::config
{'<cgi_url_base>'}\/nmiscgi.pl?type=drawgraph&node=$nod
e&graph=$graphtype&length=$graphlength&start=$start_time&end=$end_time
&width=660&height=150&interface=$interface\">";


and for logs.pl is:

475c475
<                               "<a href=\"/cgi-nmis/nmiscgi.pl?
node=$lnode\"><img alt=\"NMIS\" src=\"$NMIS::config{nmis
_icon}\" border=\"0\"></a>".
---
>                               "<a href=\"$NMIS::config
{'<cgi_url_base>'}/nmiscgi.pl?node=$lnode\"><img alt=\"NMIS\" sr
c=\"$NMIS::config{nmis_icon}\" border=\"0\"></a>".
500c500
<                       "<a href=\"/cgi-nmis/nmiscgi.pl?
node=$lnode\"><img alt=\"NMIS\" src=\"$NMIS::config{nmis_icon}\"
border=\"0\"></a>"
---
>                       "<a href=\"$NMIS::config
{'<cgi_url_base>'}/nmiscgi.pl?node=$lnode\"><img alt=\"NMIS\"
src=\"$NMI
S::config{nmis_icon}\" border=\"0\"></a>"
561c561
<                       "<a href=\"/cgi-nmis/nmiscgi.pl?
node=$lnode\"><img alt=\"NMIS Dash\" src=\"$NMIS::config{nmis_ic
on}\" border=\"0\"></a> "
---
>                       "<a href=\"$NMIS::config
{'<cgi_url_base>'}/nmiscgi.pl?node=$lnode\"><img alt=\"NMIS Dash\"
src=\
"$NMIS::config{nmis_icon}\" border=\"0\"></a> "
```

DONE with model.csv

getNodeInfo

```
elsif ( $NMIS::systemTable{sysDescr} =~ /sun/i and $NMIS::systemTable{nodeVendor} ne "Cisco Sy
        $NMIS::systemTable{nodeModel} = "SunSolaris";
        $NMIS::systemTable{nodeType} = "server";
        $NMIS::systemTable{netType} = "lan";
        $NMIS::systemTable{supported} = "true";
}

line 1534 (make older IOS generic)

if (    ( $NMIS::systemTable{nodeVendor} eq "Cisco Systems" ) &&
                                ( $NMIS::systemTable{nodeType} eq "router" ) &&
                                $NMIS::systemTable{sysDescr} !~ /Version 10.3/
                        ) {
                                $NMIS::systemTable{nodeModel} = "CiscoRouter";
                                $NMIS::systemTable{nodeType} = "router";
                                $NMIS::systemTable{netType} = "wan";
                                $NMIS::systemTable{supported} = "true";
                        }
```

NET-SNMP

Last updated 21 June 2001	Online Version	NMIS Home Page

Setting up NET-SNMP (nee UCD-SNMP)

- Download NET-SNMP or more info is available on the NET-SNMP Home Page
- Download SINS MIB Archive
- set the environment "MIBS=ALL"
- Setup the trap daemon to log to syslog or to a file.

Setting up TRAPS to be put into syslog

Create or edit the file /usr/local/share/snmp/snmptrapd.conf and add the following entry:

```
traphandle default /usr/local/bin/traplog.pl
```

Start the trap daemon and put into syslog

The command to start the NET-SNMP trap daemon and put the results in syslog:

```
export MIBS=ALL
/usr/local/sbin/snmptrapd -s -l0 -Of -c/usr/local/share/snmp/snmptrapd.conf
```

Add the following entry to syslog.conf, touch the file and restart the syslog daemon.

```
local0.debug /var/log/trap.log
```


```
                    GNU GENERAL PUBLIC LICENSE
                       Version 2, June 1991

 Copyright (C) 1989, 1991 Free Software Foundation, Inc.
                       59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                            Preamble

  The licenses for most software are designed to take away your
freedom to share and change it.  By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users.  This
```

General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

□

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to

address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

□

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) 19yy  <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
    Gnomovision version 69, Copyright (C) 19yy name of author
    Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
  Yoyodyne, Inc., hereby disclaims all copyright interest in the program
  `Gnomovision' (which makes passes at compilers) written by James Hacker.

  <signature of Ty Coon>, 1 April 1989
  Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.